Change Rate Estimation and Optimal Freshness in Web Page Crawling

Konstantin Avrachenkov [1], Kishor Patil[1], and Gugan Thoppe[2]

[1]INRIA Sophia Antipolis, France 06902*
[2]Indian Institute of Science, Bengaluru, India 560012

Abstract

For providing quick and accurate results, a search engine maintains a local snapshot of the entire web. And, to keep this local cache fresh, it employs a crawler for tracking changes across various web pages. However, finite bandwidth availability and server restrictions impose some constraints on the crawling frequency. Consequently, the ideal crawling rates are the ones that maximise the freshness of the local cache and also respect the above constraints.

Azar et al. [2] recently proposed a tractable algorithm to solve this optimisation problem. However, they assume the knowledge of the exact page change rates, which is unrealistic in practice. We address this issue here. Specifically, we provide two novel schemes for online estimation of page change rates. Both schemes only need partial information about the page change process, i.e., they only need to know if the page has changed or not since the last crawled instance. For both these schemes, we prove convergence and, also, derive their convergence rates. Finally, we provide some numerical experiments to compare the performance of our proposed estimators with the existing ones (e.g., MLE).

1 Introduction

The world wide web is gigantic: it has a lot of interconnected information and both the information and the connections keep changing. However, irrespective of the challenges arising out of this, a user always expects a search engine to instantaneously provide accurate and up-to-date results. A search engine deals with this by maintaining a local cache of all the useful web pages and their links. As the freshness of this cache determines the quality of the search results, the search engine regularly updates it by employing a crawler (also referred to as a web spider or a web robot). The job of a crawler is (a) to access various web pages at certain frequencies so as to determine if any changes have happened to the content since the last crawled instance and (b) to update the local cache whenever there is a change. To understand the detailed working of crawlers, see [13, 6, 14, 17, 12].

In general, a crawler has two constraints on how often it can access a page. The first one is due to limitations on the available bandwidth. The second one—also known as the politiness constraint—arises when a server imposes limits on the crawl frequency. The latter implies that the crawler can not access pages on that server too often in a short amount of time. Such constraints cannot be ignored, since otherwise the server may forbid the crawler from all future accesses.

In summary, to identify the ideal rates for crawling different web pages, a search engine needs to solve the following optimisation problem: Maximise the freshness of the local database subject to constraints on the crawling frequency.

*This paper has been accepted in the 13th EAI International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS '20, May 18–20, 2020, Tsukuba, Japan

*email: k.avrachenkov@inria.fr, kishor.patil@inria.fr, gugan.thoppe@gmail.com

In the early variants of this problem, the freshness of each page was assumed to be equally important [8, 12]. In such cases, experimental evidence surprisingly shows that the uniform policy—crawl all pages at the same frequency irrespective of their change rates—is more or less the optimal crawling strategy.

Starting from the pioneering work in [9], however, the freshness definition was modified to include different weights for different pages depending on their importance, e.g., represented as the frequency of requests for the pages. The motivation for this change was the fact that only a finite number of pages can be crawled in any given time frame. Hence, to improve the utility of the local database, important pages should be kept as fresh as possible. Not surprisingly, under this new definition, the optimal crawling policy does indeed depend on the page change rates. This was numerically demonstrated first in [9] for a setup with a small number of pages. A more rigorous derivation of this fact was recently given in the path breaking paper [2] by Azar et al. In fact, this work also provides a near-linear time algorithm to find a near-optimal solution.

A separate study [1, 16] provides a Whittle index based dynamic programming approach to optimise the schedule of a web crawler. In that context, the page/catalogue freshness estimate also influences the optimal crawling policy and its good estimation is needed.

Our work is mainly motivated by the work from Azar et al. [2]. In particular, input to their algorithm is the actual page change rates. However, in practice, these values are not known in advance and, instead, have to be estimated. This is the issue that we address in this paper.

Our main contributions can be summarised as follows. First, we propose two novel approaches for online estimation of the actual page change rates. The first is based on the Law of Large Numbers (LLN), while the second is derived using the Stochastic Approximation (SA) principles. Second, we theoretically show that both these estimators almost surely (a.s.) converge to the actual change rate values, i.e., both our estimators are asymptotically consistent. Furthermore, we also derive their convergence rates in the expected error sense. Finally, we provide some simulation results to compare the performance of our online schemes to each other and also to that of the (offline) MLE estimator. Alongside, we also show how our estimates can be combined with the algorithm in [2] to obtain near-optimal crawling rates.

The rest of this paper is organised as follows. The next section provides a formal summary of this work in terms of the setup, goals, and key contributions. It also gives the explicit update rules for our two online schemes. In Section 3, we discuss their convergence and converge rates and also provide the formal analysis for the same. The numerical experiments discussed above are given in Section 4. We conclude in Section 5 with some future directions.

2 Setup, Goal, and Key Contributions

The three topics are individually described below.

Setup: We assume the following. The local cache consists of copies of N pages and w_i denotes the importance of the $i-$th page. Further, each page changes independently and the actual times at which page i changes is a homogeneous Poisson point process in $[0, \infty)$ with a constant but unknown rate Δ_i. Independent of everything else, page i is crawled (accessed) at the random instances $\{t_k\}_{k\geq 0} \subset [0, \infty)$, where $t_0 = 0$ and the inter-arrival times, i.e., $\{t_k - t_{k-1}\}_{k\geq 1}$, are iid exponential random variables with a known rate p_i. Thus, the times at which page i is crawled is also a Poisson point process but with rate p_i. At time instance t_k, we get to know if page i got modified or not in the interval $(t_{k-1}, t_k]$, i.e., we can access the value of the indicator

$$I_k := \begin{cases} 1, & \text{if page i got modified in } (t_{k-1}, t_k], \\ 0, & \text{otherwise.} \end{cases}$$

We emphasise that each page is crawled independently. In other words, the notations $\{t_k\}$ and $\{I_k\}$ defined above do depend on i. However, we hide this dependence for the sake

of notational simplicity. We shall follow this practice for the other notations as well; the dependence on i should be clear from the context.

Although the above assumptions are standard in the crawling literature, nevertheless, we now provide a quick justification for the same. Our assumption that the page change process is a Poisson point process is based on the experiments reported in [4, 5, 7]. Some generalised models for the page change process have also been considered in the literature [15, 18]; however, we do not pursue these ideas here. Separately, our assumption on $\{I_k\}$ is based on the fact that a crawler can only access incomplete knowledge about the page change process. In particular, a crawler does not know when and how many times a page has changed between two crawling instances. Instead, all it can track is the status of a page at each crawling instance and know if it has changed or not with respect to the previous access. Sometimes, it is possible to also know the time at which the page was last modified [6, 10], but we do not consider this case here.

Goal: Develop online algorithms for estimating Δ_i in the above setup. Subsequently, find optimal crawling rates $\{p_i^*\}$ so that the overall freshness of the local cache defined by

$$\mathbb{E}\Big[\frac{1}{T}\int_0^T \Big(\sum_{i=1}^N w_i \mathbb{1}\{\text{Fresh}(i,t)\}\Big)dt\Big] \tag{1}$$

is maximised subject to $\sum_{i=1}^N p_i \le B$. Here, $T > 0$ is some finite horizon, $B \ge 0$ is a bound on the overall crawling frequency, $\mathbb{1}\{\}$ is the indicator, and $\text{Fresh}(i,t)$ is the event that page i is fresh at time t, i.e., the local copy matches the actual page.

Key Contributions: We present two online methods for estimating Δ_i, the first based on the LLN and the second based on SA. If $\{x_k\}$ and $\{y_k\}$ denote the iterates of these two methods, then their update rules are as shown below.

- *LLN Estimator*: For $k \ge 1$,

$$x_k = p_i \hat{I}_k/(k + \alpha_k - \hat{I}_k). \tag{2}$$

  Here, $\hat{I}_k = \sum_{j=1}^k I_j$; hence, $\hat{I}_k = \hat{I}_{k-1} + I_k$. And, $\{\alpha_k\}$ is any positive sequence satisfying the conditions in Theorem 1; e.g., $\{\alpha_k\}$ could be $\{1\}$, $\{\log k\}$, or $\{\sqrt{k}\}$.

- *SA Estimator*: For $k \ge 0$ and some initial value y_0,

$$y_{k+1} = y_k + \eta_k[I_{k+1}(y_k + p_i) - y_k]. \tag{3}$$

  Here, $\{\eta_k\}$ is any stepsize sequence that satisfies the conditions in Theorem 2. For example, $\{\eta_k\}$ could be $\{1/(k+1)^\gamma\}$ for some $\gamma \in (0, 1]$.

We call these methods online because the estimates can be updated on the fly as and when a new observation I_k becomes available. This contrasts the MLE estimator in which one needs to start the calculation from scratch each time a new data point arrives. Also, unlike MLE, our estimators are never unstable. See Section 3.3 for the complete details on this.

Our main results include the following. We show that both $\{x_k\}$ and $\{y_k\}$ converge to Δ_i a.s. Further, we show that

1. $\mathbb{E}\|x_k - \Delta_i\| = O\left(\max\left\{k^{-1/2}, \alpha_k/k\right\}\right)$, and
2. $\mathbb{E}\|y_k - \Delta_i\| = O(k^{-\gamma/2})$ if $\eta_k = (k+1)^\gamma$ with $\gamma \in (0, 1)$.

Finally, we provide three numerical experiments for judging the strength of our two estimators. In the first one, we compare the performance of our estimators to each other and also to that of the Naive estimator and the MLE estimator described in [10]. In the second one, we combine our estimates with the algorithm in [2] and compute the optimal crawling rates. Subsequently, we use this to measure the overall freshness of the local cache. In the last and final experiment, we look at the behaviour of our estimators for different choices of the sequences $\{\alpha_k\}$ and $\{\eta_k\}$.

3 Change rate estimation

Here, we provide a formal convergence and convergence rate analysis for our two estimators. Thereafter, we compare their behaviours to that of the estimators that already exist in the literature—the Naive estimator, the MLE estimator, and the Moment Matching (MM) estimator.

3.1 LLN Estimator

Our first aim here is to obtain a formula for $\mathbb{E}[I_1]$. We shall use this later to motivate the form of our LLN estimator.

Let $\tau_1 = t_1 - t_0 = t_1$. Then, as per our assumptions in Section 2, τ_1 is an exponential random variable with rate p_i. Also, $\mathbb{E}[I_1|\tau_1 = \tau] = 1 - \exp(-\Delta_i\tau)$. These two facts put together show that

$$\mathbb{E}\big[I_1\big] = \Delta_i/(\Delta_i + p_i). \tag{4}$$

This gives the desired formula for $\mathbb{E}[I_1]$.

From this last calculation, we have

$$\Delta_i = p_i\mathbb{E}[I_1]/(1 - \mathbb{E}[I_1]) \tag{5}$$

Separately, because $\{I_k\}$ is an iid sequence and $\mathbb{E}|I_1| \leq 1$, it follows from the strong law of large numbers that $\mathbb{E}\big[I_1\big] = \lim_{k\to\infty}\sum_{j=1}^{k} I_j/k$ a.s. Thus,

$$\Delta_i = p_i\frac{\lim_{k\to\infty}\sum_{j=1}^{k} I_j/k}{1 - \lim_{k\to\infty}\sum_{j=1}^{k} I_j/k} \quad \text{a.s.}$$

Consequently, a natural estimator for Δ_i is

$$x'_k = p_i\frac{\sum_{j=1}^{k} I_j/k}{1 - \sum_{j=1}^{k} I_j/k} = p_i\frac{\hat{I}_k}{k - \hat{I}_k}, \tag{6}$$

where $\hat{I}_k$ is as defined below (2).

Unfortunately, the above estimator faces an instability issue, i.e., $x'_k = \infty$ when $I_1, \ldots, I_k$ are all 1. To fix this, one can add a non-zero term in the denominator. The different choices then gives rise to the LLN estimator defined in (2).

The following result discusses the convergence and convergence rate of this estimator.

Theorem 1. *Consider the estimator given in* (2) *for some positive sequence* $\{\alpha_k\}$.

1. *If* $\lim_{k\to\infty}\alpha_k/k = 0$, *then* $\lim_{k\to\infty} x_k = \Delta_i$ *a.s.*
2. *Additionally, if* $\lim_{k\to\infty}\log(k/\alpha_k)/k = 0$, *then*

$$\mathbb{E}|x_k - \Delta_i| = O\left(\max\left\{k^{-1/2}, \alpha_k/k\right\}\right).$$

Proof. Let $\mu = \mathbb{E}[I_1]$, $\overline{I}_k = \hat{I}_k/k$, and $\overline{\alpha}_k = \alpha_k/k$. Then, observe that (2) can be rewritten as $x_k = p_i\overline{I}_k/(1+\overline{\alpha}_k-\overline{I}_k)$. Now, $\lim_{k\to\infty}\overline{I}_k = \mu$ a.s. and $\lim_{k\to\infty}\overline{\alpha}_k = 0$. The first claim holds due to the strong law of large numbers, while the second one is true due to our assumption. Statement (1) is now easy to see.

We now derive Statement (2). From (5), we have

$$|x_k - \Delta_i| = \left|x_k - p_i\frac{\mu}{1-\mu}\right| \leq p_i\left(|A_k| + |B_k|\right),$$

where

$$A_k = \frac{\overline{I}_k}{\overline{\alpha}_k + 1 - \overline{I}_k} - \frac{\mu}{\overline{\alpha}_k + 1 - \mu}$$

and

$$B_k = \frac{\mu}{\overline{\alpha}_k + 1 - \mu} - \frac{\mu}{1-\mu}.$$

Since $\alpha_k > 0$ and, hence, $\overline{\alpha}_k > 0$,

$$|B_k| = \overline{\alpha}_k \frac{\mu}{(1-\mu)(\overline{\alpha}_k + (1-\mu))} \leq \overline{\alpha}_k \frac{\mu}{(1-\mu)^2}.$$

Similarly,

$$|A_k| \leq \left(\frac{1+\overline{\alpha}_k}{1-\mu}\right)\left(\frac{|\overline{I}_k - \mu|}{\overline{\alpha}_k + 1 - \overline{I}_k}\right).$$

Because we have assumed $\overline{\alpha}_k \to 0$, we get $\lim_{k\to\infty} \mathbb{E}[B_k] = 0$. It remains to show $\lim_{k\to\infty} \mathbb{E}[A_k] = 0$. Towards that, let $\{\delta_k\}$ be a positive sequence that we will pick later. Then,

$$\mathbb{E}\left[\frac{|\overline{I}_k - \mu|}{\overline{\alpha}_k + 1 - \overline{I}_k}\right] \leq \mathbb{E}[C_k] + \mathbb{E}[D_k]$$

where

$$C_k = \frac{|\overline{I}_k - \mu|}{\overline{\alpha}_k + 1 - \overline{I}_k} \mathbb{1}\{\overline{I}_k - \mu \leq \delta_k \mu\}$$

and

$$D_k = \frac{|\overline{I}_k - \mu|}{\overline{\alpha}_k + 1 - \overline{I}_k} \mathbb{1}\{\overline{I}_k - \mu \geq \delta_k \mu\}.$$

On the one hand,

$$\mathbb{E}[C_k] \leq \frac{\mathbb{E}|\overline{I}_k - \mu|}{\overline{\alpha}_k + 1 - (1+\delta_k)\mu} \leq \frac{\sqrt{\mathrm{Var}[I_1]}}{\sqrt{k}(\overline{\alpha}_k + 1 - (1+\delta_k)\mu)}.$$

On the other hand, since $|\overline{I}_k - \mu| \leq 2$ and $1 - \overline{I}_k \geq 0$, it follows by applying the Chernoff bound that

$$\mathbb{E}[D_k] \leq \frac{2}{\overline{\alpha}_k} \Pr\{\overline{I}_k \geq (1+\delta_k)\mu\} \leq \frac{2}{\overline{\alpha}_k} \exp\left(-k\delta_k^2 \mu/3\right).$$

We now pick $\{\delta_k\}$ so that $\delta_k^2 = 6\log(1/\overline{\alpha}_k)/(k\mu)$ for all $k \geq 1$. Then, $\mathbb{E}[D_k] \leq 2\overline{\alpha}_k$. Now, due to our assumptions on $\{\alpha_k\}$, $\lim_{k\to\infty} \mathbb{E}[D_k] = 0$. Similarly, $\lim_{k\to\infty} \delta_k = 0$, whence it follows that $\lim_{k\to\infty} \mathbb{E}[C_k] = 0$. These relations together then show that $\lim_{k\to\infty} \mathbb{E}[A_k] = 0$.

The desired result now follows. □

3.2 SA Estimator

Here, we use the theory of stochastic approximation to study the behaviour of our SA estimator.

Theorem 2. *Consider the estimator given in* (3) *for some positive stepsize sequence* $\{\eta_k\}$.

1. *Suppose that* $\sum_{k=0}^{\infty} \eta_k = \infty$ *and* $\sum_{k=0}^{\infty} \eta_k^2 < \infty$. *Then,* $\lim_{k\to\infty} y_k = \Delta_i$ *a.s.*
2. *Suppose that* $\eta_k = 1/(k+1)^\gamma$ *with* $\gamma \in (0,1)$. *Then,*

$$\mathbb{E}\|y_k - \Delta_i\| = O\left(k^{-\gamma/2}\right).$$

Proof. For $k \geq 0$, let $\mathcal{F}_k := \sigma(\Delta^0, I_1, \ldots, I_k)$. Then, from (4) and the fact that $\{I_k\}$ is an iid sequence, we get

$$\mathbb{E}[I_{k+1}(y_k + p_i) - y_k | \mathcal{F}_k] = \frac{\Delta_i}{\Delta_i + p_i}(y_k + p_i) - y_k = h(y_k),$$

where $h(y) = (\Delta_i - y)p_i/(\Delta_i + p_i)$. Hence, one can rewrite (3) as

$$y_{k+1} = y_k + \eta_k[h(y_k) + M_{k+1}], \tag{7}$$

where

$$\begin{aligned} M_{k+1} &= [I_{k+1}(y_k + p_i) - y_k] - h(y_k) \\ &= \left[I_{k+1} - \frac{\Delta_i}{\Delta_i + p_i}\right](y_k + p_i). \end{aligned}$$

Since $\mathbb{E}[M_{k+1}|\mathcal{F}_k] = 0$ for all $k \geq 0$, $\{M_k\}$ is a martingale difference sequence. Consequently, (7) is a classical SA algorithm whose limiting ODE is

$$\dot{y}(t) = h(y). \tag{8}$$

Now, Statement (1) follows from Corollary 4 and Theorem 7 in Chapters 2 and 3, respectively, of [3], provided we show that:

i.) h is a globally Lipschitz continuous function.

ii.) Δ_i is an unique globally asymptotically stable equilibrium of (8).

iii.) $\sum_{k=0}^{\infty} \eta_k = \infty$ and $\sum_{k=0}^{\infty} \eta_k^2 < \infty$.

iv.) $\{M_k\}$ is a martingale difference sequence with respect to the filtration $\{\mathcal{F}_k\}$. Further, there is a constant $C \geq 0$ such that $\mathbb{E}[M_{k+1}^2|\mathcal{F}_k] \leq C(1 + y_k^2)$ a.s. for all $k \geq 0$.

v.) There exists a continuous function h_∞ such that the functions $h_c(x) := h(cx)/c$, $c \geq 1$, satisfy $h_c(x) \to h_\infty(x)$ uniformly on compact sets as $c \to \infty$.

vi.) The ODE $\dot{y}(t) = h_\infty(y)$ has origin as its unique globally asymptotically stable equilibrium.

Since h is linear, the Lipschitz continuity condition trivially holds. Separately, observe that $h(\Delta_i) = 0$; this shows that Δ_i is an equilibrium point of (8). Now, $L(y) = (y - \Delta_i)^2/2$ is a Lyapunov function for (8) with respect to Δ_i. This is because $L(y) \geq 0$, while $\nabla L(y)h(y) = -p(y - \Delta_i)^2/(p_i + \Delta_i) \leq 0$; the equality holds in both these relations if and only if $y = \Delta_i$. This shows that Δ_i is a unique globally asymptotically stable equilibrium of (8), which establishes Condition ii.).

Condition iii.) trivially holds due to our assumption about $\{\eta_k\}$. Regarding the next condition, observe that $\{M_k\}$ is indeed a martingale difference sequence. Further, $|M_{k+1}| \leq |y_k| + p_i$, whence it follows that Condition iv.) also holds.

Next, let $h_\infty(y) := -yp_i/(\Delta_i + p_i)$. Then, it is easy to see that Condition v.) trivially holds. Similarly, it is easy to see that Condition vi.) holds as well.

Statement (1) now follows, as desired.

We now sketch a proof for Statement (2). First, note that

$$y_{k+1} - \Delta_i = (1 - \lambda\eta_k)(y_k - \Delta_i) + \eta_k M_{k+1},$$

where $\lambda = p_i/(\Delta_i + p_i)$. Now, since $\mathbb{E}[M_{k+1}|\mathcal{F}_k] = 0$,

$$\mathbb{E}[(y_{k+1} - \Delta_i)^2|\mathcal{F}_k] = (1 - \lambda\eta_k)^2(y_k - \Delta_i)^2 + \eta_k^2\mathbb{E}[M_{k+1}^2|\mathcal{F}_k].$$

Recall that $\mathbb{E}[M_{k+1}^2|\mathcal{F}_k] \leq C(1 + y_k^2)$ for some constant $C \geq 0$. Using this above and then repeating all the steps from the proof of [11, Theorem 3.1] gives Statement (2), as desired. □

3.3 Comparison with Existing Estimators

As far as we know, there are three other approaches in the literature for estimating page change rates—the Naive estimator, the MLE estimator, and the MM estimator. The details about the first two estimators can be found in [10] while, for the third one, one can look at [19]. We now do a comparison, within the context of our setup, between these estimators and the ones that we have proposed.

The Naive estimator simply uses the average number of changes detected to approximate the rate at which a page changes. That is, if $\{z_k\}$ denote the values of the Naive estimator

then, in our setup, $z_k = p_i \hat{I}_k/k$, where $\hat{I}_k$ is as defined below in (2). The intuition behind this is the following. If τ_1 is as defined at the beginning of Section 3.1, then observe that $\mathbb{E}[N(\tau_1)] = \Delta_i/p_i$. Hence, the Naive estimator tries to approximate $\mathbb{E}[N(\tau_1)]$ with $\hat{I}_k/k$ so that the previous relation can then be used for guessing the change rate.

Clearly, $\mathbb{E}[z_k] = p_i\Delta_i/(\Delta_i + p_i) \neq \Delta_i$. Also, from the strong law of large numbers, $z_k \overset{a.s.}{\to} p_i\Delta_i/(\Delta_i + p_i) \neq \Delta_i$. Thus, this estimator is not consistent and is also biased. This is to be expected since this estimator does not account for all the changes that occur between two consecutive accesses.

Next, we look at the MLE estimator. Informally, this estimator identifies the parameter value that has the highest probability of producing the observed set of observations.In our setup, the value of the MLE estimator is obtained by solving the following equation for Δ_i :

$$\sum_{j=1}^{k} I_j \, \tau_j/(\exp(\Delta_i \, \tau_j) - 1) = \sum_{j=1}^{k} (1 - I_j) \, \tau_j, \tag{9}$$

where $\tau_k = t_k - t_{k-1}$ and $\{t_k\}$ is as defined in Section 2. The derivation of this relation is given in [10, Appendix C]. As mentioned in [10, Section 4], the above estimator is consistent.

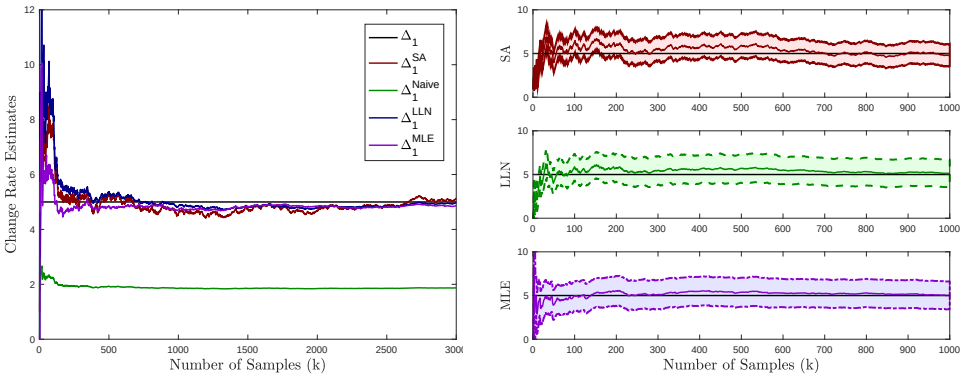
Note that the MLE estimator makes actual use of the inter-arrival crawl times $\{\tau_k\}$ unlike our two estimators and also the Naive estimator. In this sense, it fully accounts for the randomness in crawling intervals. And, as we shall see in the numerical section, the quality of the estimate obtained via MLE improves rapidly in comparison to the Naive estimator as the sample size increases.

However, MLE suffers in two aspects— computational tractability and mathematical instability. Specifically, note that the MLE estimator lacks a closed form expression. Therefore, one has to solve (9) by using numerical methods such as the Newton–Raphson method, Fisher's Scoring Method, etc. Unfortunately, using these ideas to solve (9) takes more and more time as the number of samples grow. Also note that, under the above solution ideas, the MLE estimator works in an offline fashion. In that, each time we get a new observation, (9) needs to be solved afresh. This is because there is no easy way to efficiently reuse the calculations from one iteration into the next. One reasonable alternative is to perform MLE estimation in a batch mode, i.e., wait until we gather a large number of samples and then apply one of the above-mentioned methods. However, even then the computation time will be long when k is large.

Besides the complexity, the MLE estimator is also unstable in two situations. One, when no changes have been detected ($I_j = 0$, $\forall k \in \{1, \ldots, k\}$), and the other, when all the accesses detect a change ($I_j = 1$, $\forall k \in \{1, \ldots, k\}$). In the first setting, no solution exists; in the second setting, the solution is ∞. One simple strategy to avoid these instability issues is to clip the estimate to some pre-defined range whenever one of bad observation instances occur.

Finally, we talk about the MM estimator. Here, one looks at the fraction of times no changes were detected during page accesses and, then, using a moment matching method tries to approximate the actual page change rate. In our context, the value of this estimator is obtained by solving $\sum_{j=1}^{k}(1 - I_j) = \sum_{j=1}^{k} e^{-\Delta_i \tau_j}$ for Δ_i. The details of this equation are given in [19, Section 4]. While the MM idea is indeed simpler than MLE, the associated estimation process continues to suffer from similar instability and computational issues like the ones discussed above.

We emphasise that none of our estimators suffer from any of the issues mentioned above. In particular, both our estimators are online and have a significantly simple update rule; thus, improving the estimate whenever a new data point arrives is extremely easy. Also, both our estimators are stable, i.e., the estimated values will almost surely be finite. More importantly, the performance of our estimators is comparable to that of MLE. This can be seen from the numerical experiments in Section 4.



(a) $\Delta_1 = 5, \quad p_1 = 3$ (b) 95% Confidence interval

Figure 1: Comparison between Different Estimators.

4 Numerical Results

In this section, we provide three simulations to help evaluate the strength of our estimators. In the first experiment, we look at how well our estimation ideas perform in comparison to the Naive and the MLE estimator. In the second experiment, we substitute the change rate estimates obtained via the above approaches into the algorithm given in [2] and compute the optimal crawling rates. To judge the quality of the crawling policy so obtained, we also look at the associated average freshness as defined in (1). Finally, in the third experiment, we compare the performance of our two estimators for different choices of $\{\alpha_k\}$ and $\{\eta_k\}$, respectively.

Expt. 1: Comparison of Estimation Quality

Here, we compare four different page rate estimators: LLN, SA, Naive, and MLE. Their performances can be seen in Fig 1. We now describe what is happening in the two figures there. Unless specified, the notations are as in Section 2.

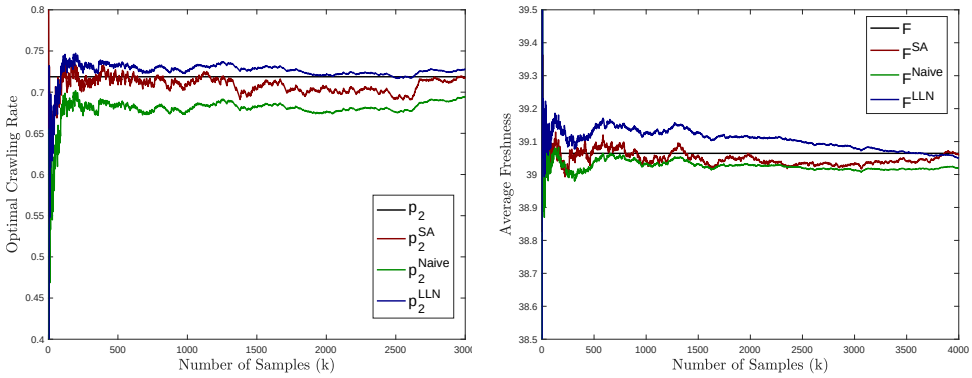
In Fig. 1(a), we work with exactly one page. We suppose that the times at which this page changes is a homogeneous Poisson point process with rate $\Delta_1 = 5$. Separately, we set the crawling frequency arbitrarily to be $p_1 = 3$. This implies that the times at which we crawl this page is another Poisson point process with rate $p_1 = 3$.

Using the above parameters, we now generate the random time instances at which this page changes. Alongside, we also sample the time instances at which this page is crawled. We then check if the page has changed or not between two successive page accesses. This generates the values of indicator sequence $\{I_k\}$.

We now give $\{I_k\}$, $\{\tau_k\}$, and p_i as input to the four different estimators mentioned above and analyse their performances. The trajectory shown in Fig. 1(a) corresponds to exactly one run of each estimator. Note that the trajectory of the estimates obtained by the SA estimator is labelled Δ_1^{SA}, etc. For the SA estimator, we had set $\eta_k = (k+1)^{-\gamma}$ with $\gamma = 0.75$. On the other hand, for our LLN estimator, we had set $\alpha_k \equiv 1$.

In Fig. 1(b), the parameter values are exactly in Fig. 1(a). However, we now run the simulation 1000 times; the page change times and the page access times are generated afresh in each run. We then look at the 95% confidence interval of the obtained estimates.

We now summarise our findings. Clearly, in each case, we can observe that performances of the MLE, LLN, and the SA estimators are comparable to each other and all of them outperform the Naive estimator. This last observation is not surprising since the Naive estimator completely ignores the missing changes between two crawling instances. However, the fact that the estimates from our approaches are close to that of the MLE estimator—both



(a) Optimal crawling rate for Page 2

(b) Average Freshness

Figure 2: Optimal Crawling Rates and Freshness

in terms of mean and variance—was indeed surprising to us. This is because, unlike MLE, our estimators completely ignore the actual lengths of the intervals between two accesses. Instead, they use p_i, which only accounts for the mean interval length.

While the plots do not show this, we once again draw attention to the fact that the time taken by each iteration in MLE rapidly grows as k increases. However, our estimators take roughly the same amount of time for each iteration.

Expt. 2: Optimal Crawling rates and Freshness

In this experiment, we consider $N = 100$ pages together. The $\{\Delta_i\}$ sequence—the mean change rates for different pages—is obtained by sampling independently from the uniform distribution on $[0, 1]$, i.e., $\Delta_i \sim U[0,\ 1]$. We further assume that the bound on the overall bandwidth is $B = 80$. The initial crawling frequencies for different pages are set by breaking up B evenly across all pages, i.e., $p_i = B/N = 0.8$ for all i. Because the p_i values are arbitrarily chosen, these are not the optimal crawling rates. We then independently generate the change and access times for each page as in Expt. 1. Subsequently, we estimate the unknown change rate for each page individually.

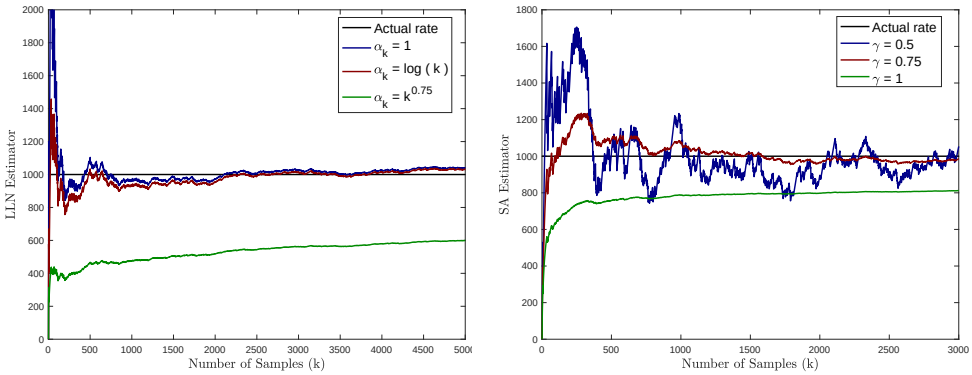
For each k, we then substitute the change rate estimates given by the different estimators into [2, Algorithm 2] and obtain the associated optimal crawling rates. In the same way, we substitute the actual Δ_i values there and obtain the true optimal crawling rates. Fig. 2(a) provides a comparison between these values for a single page. We can see that the estimate of the optimal crawling rate obtained from our approaches is much better than that of the Naive estimator.

To check how good our estimate of the true optimal crawling policy is, we look at the associated average freshness given by[1]

$$F(p) = \sum_{i=0}^{N} \frac{w_i p_i}{p_i + \Delta_i} \tag{10}$$

and compare the same to that of the true optimal crawling policy. This comparison is given in Fig. 2(b). Somewhat surprisingly, the average freshness does not vary much for all the three estimators. However, eventually, the average freshness captured by our estimators becomes much closer to the true optimal average freshness.

[1]In [2], it was shown that maximising (1) under a bandwidth constraint for large enough T corresponds to maximising (10) under the same bandwidth constraint.



(a) LLN estimator for different $\{\alpha_k\}$ choices (b) SA estimator with $\eta_k = (k+1)^{-\gamma}$for different γ choices

Figure 3: Impact of $\{\alpha_k\}$ and $\{\eta_k\}$ choices on Performance.

Expt. 3: Impact of $\{\alpha_k\}$ and $\{\eta_k\}$ choices

The theoretical results presented in Section 3 showed that the convergence rate of our estimators is affected by the choice of $\{\alpha_k\}$ and $\{\eta_k\}$, respectively. Figures 3(a) and 3(b) provide a numerical verification of the same.

The details are as follows. Here, again, we restrict our attention to one single page. For Fig. 3(a), we chose $\Delta = 1000$ and $p = 200$. Notice that the page change rate is very high, whereas the crawling frequency is relatively a low value. We then used the LLN estimator with three different choices of $\{\alpha_k\}$; these choices are shown in the figure itself. The LLN estimator with $\alpha_k = k^{0.75}$ has the worst performance. This behaviour matches the prediction made by Theorem 1.

In Fig. 3(b), we again consider the same setup as above. However, this time we run the SA estimator with three different choices of $\{\eta_k\}$; the choices are given in the figure itself. We see that the performance for $\gamma = 0.75$ is better than the $\gamma = 0.5$ case. This is as predicted in Theorem 2. However, it worsens for the $\gamma = 1$ case. Notice that the latter case is not covered by Theorem 2.

5 Conclusion and Future work

We proposed two new online approaches for estimating the rate of change of web pages. Both these estimators are computationally efficient in comparison to the MLE estimator. We first provide theoretical analysis on the convergence of our estimators and then provide numerical simulations to compare their performance with the existing estimators in the literature. From numerical experiments, we have verified that the proposed estimators perform significantly better than the Naive estimator and have extremely simple update rules which make them computationally attractive.

The performance of both our estimators currently depend on the choice of $\{\alpha_k\}$ and $\{\eta_k\}$, respectively. One aspect to analyse in the future would be to ask what would be the ideal choice for these sequences that would help attain the fastest convergence rate. Another interesting research direction is to combine the online estimation with dynamic optimisation.

Acknowledgement

This work is partly supported by ANSWER project PIA FSN2 (P15 9564-266178 \DOS0060094) and DST-Inria project "Machine Learning for Network Analytics" IFC/DST-Inria-2016-01/448.

References

[1] Avrachenkov, K. E., and Borkar, V. S. Whittle index policy for crawling ephemeral content. *IEEE Transactions on Control of Network Systems 5*, 1 (2016), 446–455.

[2] Azar, Y., Horvitz, E., Lubetzky, E., Peres, Y., and Shahaf, D. Tractable near-optimal policies for crawling. *Proceedings of the National Academy of Sciences 115*, 32 (2018), 8099–8103.

[3] Borkar, V. S. *Stochastic approximation: a dynamical systems viewpoint*, vol. 48. Springer, India, 2009.

[4] Brewington, B. E., and Cybenko, G. How dynamic is the web? *Computer Networks 33*, 1-6 (2000), 257–276.

[5] Brewington, B. E., and Cybenko, G. Keeping up with the changing web. *Computer 33*, 5 (2000), 52–58.

[6] Castillo, C. Effective web crawling. In *Acm sigir forum* (New York, NY, USA, 2005), vol. 39, Acm New York, NY, USA, Association for Computing Machinery, pp. 55–56.

[7] Cho, J., and Garcia-Molina, H. The evolution of the web and implications for an incremental crawler. In *26th International Conference on Very Large Databases* (San Francisco, CA, USA, 2000), Morgan Kaufmann Publishers Inc., pp. 1–18.

[8] Cho, J., and Garcia-Molina, H. Synchronizing a database to improve freshness. *ACM sigmod record 29*, 2 (2000), 117–128.

[9] Cho, J., and Garcia-Molina, H. Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems (TODS) 28*, 4 (2003), 390–426.

[10] Cho, J., and Garcia-Molina, H. Estimating frequency of change. *ACM Transactions on Internet Technology (TOIT) 3*, 3 (2003), 256–290.

[11] Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. Finite sample analyses for td (0) with function approximation. In *Thirty-Second AAAI Conference on Artificial Intelligence* (San Francisco, CA, USA, 2018), AAAI Press, pp. 6144–6160.

[12] Edwards, J., McCurley, K., and Tomlin, J. An adaptive model for optimizing performance of an incremental web crawler. In *Proceedings of the 10th International Conference on World Wide Web* (New York, NY, USA, 2001), vol. 8, Association for Computing Machinery, p. 106–113.

[13] Heydon, A., and Najork, M. Mercator: A scalable, extensible web crawler. *World Wide Web 2*, 4 (1999), 219–229.

[14] Kumar, R., Jain, A., and Agrawal, C. A survey of web crawling algorithms. *Advances in vision computing: An international journal 3* (2016), 1–7.

[15] Matloff, N. Estimation of internet file-access/modification rates from indirect data. *ACM Transactions on Modeling and Computer Simulation (TOMACS) 15*, 3 (2005), 233–253.

[16] Niño-Mora, J. A dynamic page-refresh index policy for web crawlers. In *Analytical and Stochastic Modeling Techniques and Applications* (Cham, 2014), Springer International Publishing, pp. 46–60.

[17] Olston, C., Najork, M., et al. Web crawling. *Foundations and Trends® in Information Retrieval 4*, 3 (2010), 175–246.

[18] Singh, S. R. Estimating the rate of web page updates. In *Proc. International Joint Conferences on Artificial Intelligence* (San Francisco, CA, USA, 2007), ACM, pp. 2874–2879.

[19] Upadhyay, U., Busa-Fekete, R., Kotlowski, W., Pal, D., and Szorenyi, B. Learning to crawl. In *Thirty-fourth AAAI Conference on Artificial Intelligence* (New York, NY, USA, 2020), AAAI press, pp. 8471–8478.